

Absolvování individuální odborné praxe

Individual Professional Practice in the Company

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 3. května 2011

.....

Abstrakt

Odborná praxe ve firmě EVici webdesign s.r.o. Firma navrhuje a vytváří internetové stránky, jedná se především o online internetové obchody. Nyní probíhá vývoj systému, který bude umožňovat snadněji vytvářet a spravovat internetové obchody. Systém je vyvíjen v programovacím jazyce PHP. Bude také umožňovat efektivněji modifikovat zdrojový kód aplikace, který bude stejný pro všechny obchody. Obchody se budou od sebe lišit pouze vzhledem, způsob administrace zůstane stejný.

Klíčová slova: e-shop, internetový obchod, administrace systému, moduly systému

Abstract

Professional practice in the company EVici webdesign s.r.o. Company designs and creates web pages, primary creates online e-shops. Nowadays the system is being developed. System will allow easier creation and management of online shops. The system is developed in PHP programming language. It will also allow more efficient modification of the source code of application, that will be the same for all shops. Shops will only differ from each other in appearance, the way of administration remains the same.

Keywords: e-shop, web shop, system administration, system modules

Seznam použitých zkratk a symbolů

AJAX	– Asynchronous JavaScript and XML – asynchronní JavaScript a XML
CSS	– Cascading Style Sheets – kaskádové styly
HTML	– HyperText Markup Language – značkový jazyk pro hypertext
JS	– JavaScript
JSON	– JavaScript Object Notation – JavaScriptový objektový zápis
PHP	– Hypertext Preprocessor – hypertextový preprocesor
SQL	– Structured Query Language – strukturovaný dotazovací jazyk

Obsah

1	Popis odborného zaměření firmy	4
2	Používané frameworky a knihovny	5
2.1	Nette Framework	5
2.2	Dibi	5
2.3	jQuery	5
2.4	JSColor	5
3	Seznam úkolů zadaných studentovi v průběhu odborné praxe	6
4	Postup řešení zadaných úkolů	7
4.1	Kategorie CRM	7
4.2	Nastavení e-shopu	9
4.3	Vytváření a úprava produktů	16
5	Teoretické a praktické znalosti	22
6	Scházející znalosti a dovednosti	23
7	Dosažené výsledky	24
8	Reference	25

Seznam obrázků

1	Hlavní menu administrace systému	8
2	Odpověď na zprávu	8
3	Záložky modulu Kontakty	9
4	Vyskakovací okno s formulářem pro úpravu měny	12
5	Dialogové okénko pro výběr barvy	12
6	Tabulka se všemi dostupnostmi	13
7	Číselné řady	14
8	Tabulka se výpisem několika produktů	15
9	Vstupní políčka pro datum s indikací platnosti	17
10	Tabulka pro editaci jednotlivých variant	18
11	Zaškrťovací políčka pro generování variant	19
12	Tabulka souvisejících produktů s posuvníky	20
13	Formulář pro úpravu metráže	21

Seznam výpisů zdrojového kódu

1	jQuery metoda pro vyskakovací okno	7
2	Ukázka kódu s šablonovým makrem pro automatické vykreslení celého formuláře	10
3	Ukázka kódu s ručním vykreslením prvků formuláře v šablonách	10
4	SQL dotaz pro výpis všech produktů do tabulky	16
5	Regulární výraz pro kontrolu data	17

1 Popis odborného zaměření firmy

Firma EVici webdesign s.r.o. (viz [1]) se zabývá návrhem internetových stránek, tvorbou e-shopů a momentálně i vývojem systému pro správu e-shopů. Tento systém bude umožňovat rychleji a snadněji vytvářet a spravovat internetové obchody nabízené klientům. Doposud se novému zákazníkovi upravil některý ze stávajících obchodů. Vytvořil se nový grafický návrh a upravil se zdrojový kód podle potřeb klienta. Nový systém bude mít jednodušší a přehlednější administraci a vlastní jádro systému bude pro všechny klienty stejné. Zákazník si nechá vypracovat pouze grafický návrh (buď firmou EVici nebo externí firmou) a ten se jednoduše pomocí šablonových maker nakóduje do funkčního internetového obchodu.

Administrace se skládá z několika desítek rozšiřujících modulů, pomocí kterých lze systém spravovat a udržovat jej stále aktuální. Participoval jsem na návrhu a vývoji těchto rozšiřujících modulů.

2 Používané frameworky a knihovny

2.1 Nette Framework

Stránky projektu [4] označují Nette framework jako výkonný framework pro pohodlné a rychlé vytváření kvalitních a moderních webových aplikací v PHP 5. Zaměřuje se na eliminaci bezpečnostních rizik při tvorbě formulářů. Původním autorem Nette Frameworku je David Grudl, o jeho další rozvoj se stará organizace Nette Foundation.

2.2 Dibi

V [5] se uvádí, že Dibi je PHP databázová vrstva, která se snaží zjednodušit zápis SQL příkazů a ulehčit rutiny, se kterými se programátor běžně setkává. Dibi má implementovanou ochranu proti SQL injection a je napsána v objektovém PHP 5.

2.3 jQuery

V [2] se uvádí jQuery jako JavaScriptová knihovna, která klade důraz na interakci mezi JavaScriptem a HTML. JQuery je určena ke změně způsobu, jakým se píše JavaScript.

2.4 JSColor

Oficiální stránky [8] projektu označují JSColor jako jednoduchou a uživatelsky přívětivou knihovnu pro výběr barvy v HTML formulářích. Rozšiřuje všechny požadované HTML políčka `<input>` o barevný dialog pro výběr barvy.

3 Seznam úkolů zadaných studentovi v průběhu odborné praxe

Firmou navržený a vyvíjený systém se skládá ze tří částí. Každá část je jistým způsobem závislá na ostatních a jedna bez druhé nedokáže správně fungovat.

- První částí je hlavní administrace, kde jsou informace o všech uživateli, klientských účtech, nastavení celého systému, nastavení jazyků a jednotlivých překladů.
- Druhou částí je administrace klientských webů, kam má přístup jen majitel obchodu a jím pověřené osoby. Tato část je velice důležitá pro správu a nastavení jednotlivých stránek, článků, produktů, rolí nebo i uživatelských práv.
- Třetí částí je samotný web, který reprezentuje firmu klienta, zobrazuje produkty (v případě e-shopu) nebo pomáhá komunikovat s klienty. Celé stránky se vykreslují podle nastavení a vyplněných údajů v administraci.

Úkoly, na kterých jsem se podílel, byly návrh a vývoj modulů pro administraci zákaznických webů. Nejdůležitější částí administrace je hlavní menu (viz Obrázek 1), které tvoří kostru celé administrace. Menu je rozděleno na několik hlavních kategorií, ty obsahují vždy další soubor kategorií (podkategorie). Jednodušší podkategorie (moduly systému) jsem dokázal zvládnout za méně než 6 hodin. Na nejsložitějších modulech jsem pracoval někdy i 3 dny, s rozpočítáním na hodiny je to přibližně 24 hodin.

4 Postup řešení zadaných úkolů

4.1 Kategorie CRM

Kategorie pro řízení vztahů se zákazníky (Customer Relationship Management) obsahuje dva důležité moduly pro vztahy se zákazníky. Na obou modulech jsem se podílel návrhem a implementací. Prvním modulem je „Komunikace“, kde se shromažďují veškeré zprávy při komunikaci se zákazníky. Druhým je modul „Kontakty“, ve kterém se spravují zákazníci nebo pouze kontakty na osoby. Obě kategorie lze vidět na obrázku 1 zařazené do hlavního menu.

4.1.1 Komunikace

První část, na které jsem pracoval, byl komunikační modul, který má na starost komunikaci provozovatele a zaměstnanců obchodu se zákazníky. Tato podkategorie je součástí CRM modulu. Modul vypisuje všechny poznámky, připomínky, vzkazy zapsané a odeslané z různých formulářů na webu (například formulář „Kontaktujte nás“) a zprostředkovává emailovou komunikaci se zákazníky. Pro lepší přehlednost jsou nejdůležitější informace o vzkazu zobrazeny na úvodní stránce modulu. Zobrazeny jsou v tabulce, datagridu.

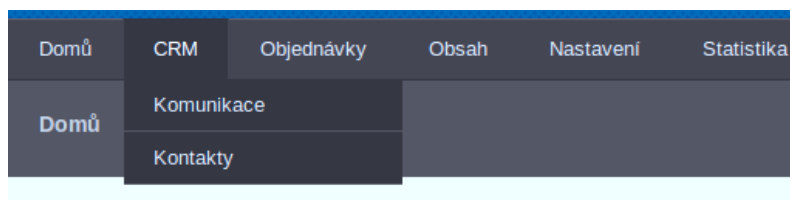
Datagrid je přídatná komponenta frameworku Nette. Lze ji najít na oficiálních stránkách [4] frameworku. V systému je často používána, protože umožňuje rychlý výpis do tabulky s následným formátováním a stránkováním řádků a sloupců. Umožňuje jedním SQL dotazem tabulku naplnit, automaticky vytvoří tlačítka pro řazení záznamů, políčka pro filtrování, hromadnou úpravu a lze přidat i tlačítka pro různé akce na jednotlivých záznamech, jako je editace nebo mazání. Důležitou součástí komponenty je možnost stránkování záznamů.

Na úvodní stránce modulu „Komunikace“ je v tabulce vypsáno několik podstatných informací o zprávě, kde podrobnější informace jsou schovány v detailu. Mezi nejdůležitější kolonky tabulky patří jméno nebo email zákazníka, předmět zprávy, z jaké stránky a z jaké komponenty byla zpráva odeslána (kontaktní formulář, diskuze k produktu, atd.). Detail zprávy je zobrazen po kliknutí na tlačítko „Detail“ (Obrázek 2).

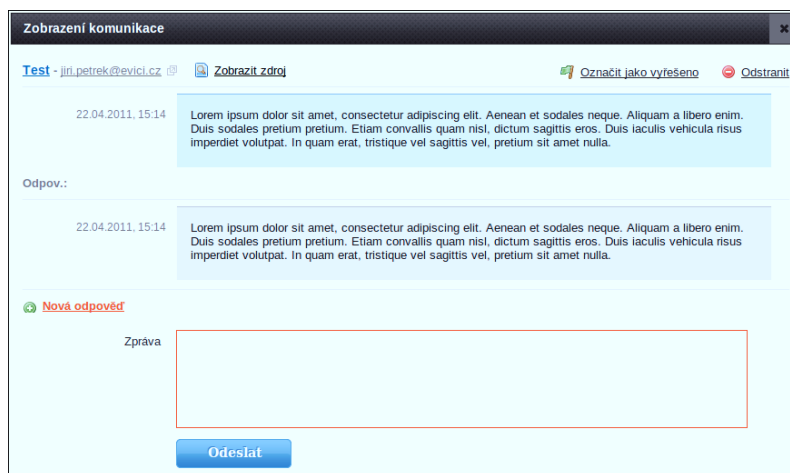
```
function dialogAddVat() {
    $('#dialogAddVat').dialog({
        title : "Přidat DPH",
        modal: true,
        resizable: false,
        draggable: true,
        width: 500,
        height: 'auto'
    });
}
```

Výpis 1: jQuery metoda pro vyskakovací okno

Podrobnější informace jsou vypisovány do vyskakovacího okna (Obrázek 2) vytvořeného jednoduchou metodou knihovny jQuery (viz Výpis 1). Okno se nezobrazuje v nové



Obrázek 1: Hlavní menu administrace systému

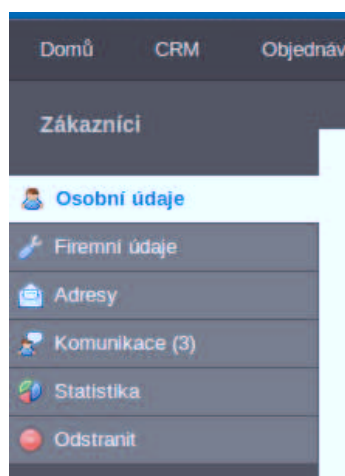


Obrázek 2: Odpověď na zprávu

instanci prohlížeče, ale jako součást aktuální webové stránky. Detail zprávy (Obrázek 2) obsahuje všechny informace o zprávě, jako je jméno odesílající osoby (pokud provedla odeslání zprávy pod přihlášením), email, datum a čas, předmět a text zprávy. Pod ní jsou vypsané všechny odpovědi, které patří k příslušné zprávě. Kliknutím na tlačítko „Nová odpověď“ vyjede JavaScriptovou animací jednoduchý formulář pro rychlou odpověď s políčkem pro text zprávy. Ikonka „Odstranit“ pro odstranění zprávy je umístěna jak v tabulce na úvodní stránce modulu, tak i v detailu. Kliknutím na kterékoliv tlačítko „Odstranit“ se spouští PHP skript, který pošle SQL dotaz na databázi a ten provede smazání záznamu v databázi. Po smazání se celá stránka znovu načte.

4.1.2 Kontakty

Další část, na které jsem se podílel, je podkategorie „Kontakty“, která se velice podobá předchozí podkategorii „Komunikace“. Úvodní stránka je tvořena tabulkou s výpisem uživatelů. Obsah tabulky s kontakty je ve výchozím stavu tříděn podle typu kontaktu (zda se jedná o zákazníka nebo jen o obyčejný kontakt). Třídění provádí SQL dotaz při načítání záznamů z databáze. Do dotazu je přidána klauzule „Order by“ a databáze všechny záznamy seřídí podle zadaného kritéria.



Obrázek 3: Záložky modulu Kontakty

Oproti podkategorii „Komunikace“ je zde jiný způsob pro úpravu záznamů. Úprava již není ve vyskakovacím okně, ale na nové stránce. Editaci jsem musel vložit na novou stránku, protože je zde příliš mnoho údajů, které by se do vyskakovacího okna nevešly.

Záložky (Obrázek 3) „Osobní údaje“, „Firemní údaje“ i záložka „Adresy“ jsou vytvořeny podobným způsobem. Všechny jsou založeny na formuláři vytvořeného pomocí frameworku Nette [4] a v šabloně vykreslovány ručně (viz Výpis 3). Ručně proto, protože v šabloně jsou prvky od sebe graficky odděleny. Toto oddělení automatické vykreslení neumožňuje (viz Výpis 2).

4.2 Nastavení e-shopu

Každý program má dnes možnost nastavení. Ne jinak je tomu i v případě internetových obchodů. Systém firmy EVici webdesign s.r.o. [1] taky tento modul obsahuje. Má ho hlavní administrace systému a mají ho i jednotlivé administrace e-shopů. Na většině podmodulů v nastavení e-shopu jsem se aktivně podílel.

4.2.1 Hlavní nastavení

První podkategorie záložky „Nastavení“ se jmenuje „Hlavní nastavení“ a nabízí globální nastavení pro zákazníkův systém. Klient si v něm může nastavit základní informace, zda je plátcem DPH, zda jsou ukládané ceny s DPH, dobu trvání statusu „Nový“ u produktu a mnoho dalšího.

Modul jsem vytvořil jedním formulářem, který je graficky rozdělen. V šabloně to znamená ručně vykreslit formulářové prvky (viz Výpis 3). Rozdělení na nastavení obchodu, nastavení produktů a nastavení uživatele je pouze z estetického hlediska pro lepší čitelnost a jasné oddělení nesouvisejících formulářových políček.

Při odeslání formuláře je potřeba informace vhodným způsobem rozkouskovat a uložit do tabulky „config“ v databázi. Do tabulky „config“ se ukládá jiným způsobem než do ostatních tabulek, protože jeden záznam reprezentuje jednu hodnotu uloženého nastavení. U všech ostatních tabulek obsahuje jeden záznam více hodnot.

Rozsekání odeslaných dat provádím uložením vždy jen jednoho prvku formuláře do dočasné proměnné a podle názvu prvku aktualizuji SQL dotazem záznam v databázi. Tímto způsobem je potřeba projít celý formulář a aktualizovat všechny záznamy.

4.2.2 Provozovatel obchodu

V této kategorii si majitel obchodu ukládá informace o provozovatelích obchodů v jednotlivých státech, ve kterých podniká. Majitel může informace o provozovatelích obchodů vypsat jen pro některé státy, přičemž stát, který si zvolil při zakládání e-shopu jako výchozí, musí mít všechny položky povinně vyplněny. Ve státech, ve kterých informace vyplněny nebudou, budou informace doplňovány z prvního (výchozího) státu. Doplňování chybějících informací o provozovatelích je potřeba vyřešit až na front-endu.

Framework Nette [4] umožňuje vytvářet kontejnery pro shlukování formulářových prvků. Touto funkcí jsou dělány všechny formuláře, kde je potřeba mít vyplněny položky ve více jazycích. Nette vytvoří kontejner s názvem jazyku a do něj vloží potřebné políčka. Funkce umožňuje vytvořit prvek se stejným názvem vícekrát.

Formulářová políčka jsou pomocí Nette frameworku vytvořena v cyklu. Ten prochází všechny aktivované jazyky a pro každý vytváří nový kontejner pro prvky „Společnost“, „Kontaktní osoba“, „Komunikace“ a „Bankovní účet“.

I zde jsem musel využít ručního vykreslení prvků formuláře do šablony (viz Výpis 3). Automatický způsob (viz Výpis 2) totiž vykresloval prvek formuláře ve všech jazycích hned pod sebe. Ale zadání mého úkolu bylo odlišné a prvky se měly vypisovat vždy v oblasti pro určitý jazyk. Z tohoto důvodu jsem požil ruční výpis (viz Výpis 3).

```
<div class="basic-form">
  {widget shopForm}
</div>
```

Výpis 2: Ukázka kódu s šablonovým makrem pro automatické vykreslení celého formuláře

```
<div class="basic-form">
  {$presenter['shopForm']->render('begin')}
  {foreach $languages as $lang}
    <fieldset>
      <legend>{_'Company'}</legend>
      <table>
        <tr class="lang-{$lang}-{if !$iterator->isFirst()}-required{/if}">
          <th c>
            {$presenter['shopForm'][$lang]['company']->label}
          </th>
          <td>
            {$presenter['shopForm'][$lang]['company']->control}
          </td>
        </tr>
      </table>
    </fieldset>
  </foreach>
</div>
```

```

    </tr>
  </table>
</fieldset>
{foreach}
{$presenter['shopForm']->render('end')}
</div>

```

Výpis 3: Ukázka kódu s ručním vykreslením prvků formuláře v šablonách

4.2.3 Nastavení DPH

Nastavení DPH je velice důležité v podnikání. V každém státě se liší a v každém státě jich může být několik. Toto je velice jednoduchý modul, který jsem podle požadavků několikrát předělával. Původně mělo každé DPH svůj název a pro každou zemi si uživatel vypsal jinou hodnotu. Nyní má každé DPH jen svou hodnotu a celé DPH je přiděleno do konkrétního státu jako výchozí nastavení pro DPH. Původně byl u každého řádku s položkou DPH posuvník pro řazení. I ten byl nakonec odstraněn a položky se řadí jen podle hodnot. V systému musí vždy existovat alespoň jedno DPH, proto jsme vytvořili jednu položku nesmazatelnou.

Výchozí nesmazatelné položky se vytváří jednoduše. Při vykreslování jednotlivých řádků tabulky se jednoduchou podmínkou testuje, zda má záznam hodnotu „default“ nastavenou. Pokud ano, řádku se přiřadí kaskádový styl „main_page_row“ a odstraní se tlačítko pro mazání. Kaskádový styl se postará o vykreslení řádku jinou barvou.

4.2.4 Měny

Úvodní stránka tohoto modulu nabízí tabulku s výpisem všech již vytvořených měn, kde první řádek nelze smazat, pouze editovat. Modul je zajímavý tím, co všechno se může u měny nastavit. Sám jsem netušil, kolik je možností. Pro systém budou nastavitelné tyto možnosti: název měny, značka měny (většinou bývá tří písmenná, např. CZK), dále levý a pravý symbol (např. \$, Kč), kurz měny, zaokrouhlovací metoda (matematicky, nahoru, dolů), zaokrouhlování na tisíce, desetiny, padesátníky nebo koruny. A nakonec nastavení pro oddělovač desetin a tisíců.

Úprava i vytváření nové měny se zobrazují ve vyskakovacím okně (Obrázek 4). Okno je vytvořeno jQuery metodou (viz Výpis 1). Všechna formulářová políčka jsou do šablony vypsána ručně (viz Výpis 3), kvůli jinému rozložení prvků.

4.2.5 Stavy objednávek

Do podkategorie se stavy objednávek jsem přinesl vlastní nápad. Zadání bylo, aby jednotlivé stavy byly barevně rozlišeny. Protože by uživatel musel psát barvu v hexadecimálním znakovém kódu barvy, byl můj nápad zobrazovat barvu v plovoucím okně nad prvkem (Obrázek 5), aby si ji uživatel mohl jednoduše vybrat kliknutím myši. Toto nabízí knihovna jsColor [8]. Lze ji kaskádovým stylem „color“ navázat na libovolný formulářový prvek. O dialog nad prvkem se postará knihovna.

Upravit měnu

Aktivní ☒

Název

Kód měny

Kurz měny

Levý symbol	Kurz	Pravý symbol
<input type="text"/>	<input type="text" value="17.67"/>	<input type="text" value="Kč"/>

Zaokrouhlovací metoda

Zaokrouhlování na

Oddělovač desetín

Oddělovač tisíců

Obrázek 4: Vyskakovací okno s formulářem pro úpravu měny

Upravit stav

Název EN

CZ

DE

Barva

Obrázek 5: Dialogové okénko pro výběr barvy

Název	Skladový limit
on request	
not available	0 jednotek
available	1 - 3 jednotek
to 5 days	4 - 9 jednotek
to 10 days	10 a více jednotek

Obrázek 6: Tabulka se všemi dostupnostmi

4.2.6 Jednotky

Další velice jednoduchá podkategorie je „Jednotky“. Ta vypisuje do tabulky vytvořené jednotky. Při úpravě jednotky vyskočí jQuery metodou (viz Výpis 1) okno pro úpravu názvu jednotky ve všech dostupných jazycích. I když je to nejjednodušší modul, na kterém jsem pracoval, našli jsme spolu s ostatními programátory jednu chybu, spojenou s jazyky. Pokud máme jednotku například v češtině a v angličtině, zobrazuje se formulář jen pro tyto dva jazyky. V případě, kdy si zákazník objedná do celého systému další jazyk, například němčinu, přestává úprava jednotek fungovat, protože aktualizují jen dva vytvořené záznamy v databázi, ve kterých je jednotka uložena. Třetí záznam pro němčinu se nevytvoří. Řešení problému je provést kontrolu existence záznamu v databázi a na jejím základě záznam buď vytvořit, nebo pouze upravit.

4.2.7 Dostupnost zboží

Na rozdíl od stavu objednávek zde nevypisují dostupnost v barvičkách, ale dopočítává se zde dostupnost zboží (Obrázek 6). Modul je tvořen přehlednou tabulkou, kde jsou přednastaveny tři hodnoty, které nelze smazat (na dotaz, nedostupné, dostupné). Další si může zákazník vytvořit. Mimo titulku musí být vyplněna i hodnota „od“ kolika jednotek bude tato dostupnost na produkt aplikována. Otázkou k zamyšlení bylo, zda ukládat hodnoty „od“ nebo „do“ počtu jednotek. V případě použití „od“ využíváme hodnotu „NULL“ pro položku „na dotaz“, hodnotu „nula“ jako „nedostupné“ a hodnoty větší než nula jsou brány jako zboží „dostupné“. V opačném případě by jako hodnota „nekonečno“ muselo být uloženo obrovské číslo nebo hodnota „NULL“. Tím bychom ztratili jednu hodnotu pro položku „na dotaz“.

Při výpisu tabulky (Obrázek 6) s dostupnostmi zboží, jsem využil PHP metody „next“, která obsahuje ukazatel na další políčko v poli. Tím lze jednoduše docílit výstupu „1 – 3

Objednávky	O	156	1	» O1561
Zákazníci	C	0000002003	&	» C0000002003&
Faktury	I	000005	*	» I000005*
<input type="button" value="Uložit"/>				

Obrázek 7: Číselné řady

jednotek“, kde hodnota „1“ je ukazatel na aktuální prvek v poli a hodnota „3“ ukazatel na následující. Pokud ukazatel na následující prvek přesahuje rozměry pole, tzn., že prvek je poslední, vypíše jen „10 a více jednotek“.

4.2.8 Číselné řady

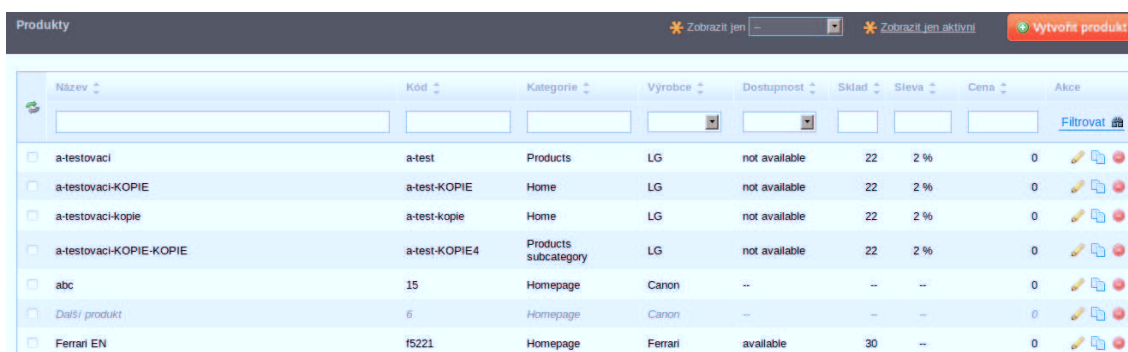
Podkategorie „Číselné řady“ (Obrázek 7) slouží majitelům nebo provozovatelům e-shopů pro číslování objednávek, faktur a zákazníků. V modulu si mohou nastavit, jaký formát bude řada mít. Řady mohou mít svůj prefix i sufix. Nejdůležitější hodnotou u číselné řady je číslo, kterému se inkrementuje hodnota při vytvoření nové objednávky, faktury nebo zákazníka. Každé číslo pak může být uvozeno nulami, to jen pro lepší čitelnost. Výsledek pak může vypadat takto: „Z00099-11“, následující bude „Z00100-11“, kde „Z“ je prefix, „00099“ samotné číselné počítadlo a „-11“ libovolný sufix.

S prefixem a sufixem není žádný problém při ukládání do databáze. Problém nastal při řešení způsobu ukládání čísla, které se bude vždy inkrementovat. Nejdříve si spočítám délku řetězce čísla uvozeného nulami. Pak v cyklu projedu všechny znaky řetězce, dokud nenarazím na první nenulový znak a nuly z řetězce vymažu. Odděleně tak uložím do databáze délku řetězce a výsledné číslo bez úvodních nul. Při zpětném dopočítávání doplním nuly zpět před číslo, aby klient nepoznal žádný rozdíl v uložení do databáze.

4.2.9 Doprava a platba

Modul se skládá ze dvou záložek. První je pro nastavení plateb, druhá pro nastavení dopravy. Nejdříve bylo potřeba udělat modul pro platbu, protože modul „Doprava“ obsahuje informace o platbách.

Modul „Platby“ se od ostatních modulů mírně liší. Nepoužívá totiž tlačítko pro přidání nebo pro odebrání některé z položek v seznamu. Položky jsou do seznamu vloženy hlavním administrátorem a nikdo nemá práva je libovolně měnit. Jediné co zákazník může, je vybrat si, jestli bude konkrétní platba pro jeho e-shop aktivní či naopak. V základní verzi systému budou pouze tři druhy plateb a to: hotově, dobírkou a převodem. Později přibudou další např. platba kartou nebo úvěr. Všechny nápisy a popisky jsou řešeny ve všech jazykových mutacích.



Název	Kód	Kategorie	Výrobce	Dostupnost	Sklad	Sleva	Cena	Akce
<input type="checkbox"/> a-testovací	a-test	Products	LG	not available	22	2 %	0	
<input type="checkbox"/> a-testovací-KOPIE	a-test-KOPIE	Home	LG	not available	22	2 %	0	
<input type="checkbox"/> a-testovací-kopie	a-test-kopie	Home	LG	not available	22	2 %	0	
<input type="checkbox"/> a-testovací-KOPIE-KOPIE	a-test-KOPIE4	Products subcategory	LG	not available	22	2 %	0	
<input type="checkbox"/> abc	15	Homepage	Canon	--	--	--	0	
<input type="checkbox"/> Další produkt	6	Homepage	Canon	--	--	--	0	
<input type="checkbox"/> Ferrari EN	f5221	Homepage	Ferrari	available	30	--	0	

Obrázek 8: Tabulka se výpisem několika produktů

Problém, na který jsem v průběhu vytváření modulu narazil, spočíval v ukládání DPH (daně z přidané hodnoty). V první fázi byl modul navržen pro ukládání jedné ceny a jednoho DPH pro všechny státy stejně. Tento způsob je však v praxi nepoužitelný, proto jsem musel upravit zdrojové kódy PHP skriptů a vytvořit tak pro každý stát vlastní položku pro cenu a vlastní položku pro DPH. Hodnoty pro položku DPH jsou vybírány z předcházejícího modulu „Nastavení DPH“.

Po ukončení prací na modulu „Platby“ jsem mohl přikročit k dokončení modulu „Doprava“. Modul „Doprava“ má různé nápisy a popisky, které je potřeba evidovat ve všech jazycích. Na modulu jsem strávil, v poměru s ostatními, nejvíce času. Důvod, proč mi trvalo vytvořit tento modul tak dlouho, byl ve výpisu seznamu plateb a výpisu seznamu států. Správce systému, který bude upravovat dopravu, by měl být schopen zaškrtnout si jen platební metody a státy, určené pro konkrétní dopravu. Například doprava poštou může být placena jen hotově nebo dobírkou, ale doprava přepravní službou PPL může být placena hotově, převodem i dobírkou. To všechno například pro dva státy, Česko a Slovensko.

Nette kontejnerů jsem využil i v případě vytváření států a platebních metod. I když kontejnery práci o moc neurychlí, výsledný kód je lépe čitelný. Seznam států byl takový programátorský oříšek, který bylo potřeba rozlousknout. Problém byl v označování správných států, kde státy patřící k dopravě vytáhneme z vazební databázové tabulky. Původně byl seznam zobrazován ve vyskakovacím okně. Z nápadu ale sešlo kvůli složitějšímu kódu a složitější implementaci. Seznam států nakonec zůstal ve stejném vyskakovacím okně metodou jQuery (viz Výpis 1), které se zobrazuje pro vkládání informací o dopravě. Pro všechny státy jsem využil Nette kontejner, do kterého všechny státy při vytváření komponenty nahraji. Formulář se státy však přesahoval velikost vyskakovacího jQuery okna. Nezbyvalo nic jiného než využít kaskádového stylu „overflow: auto“, který vytvořil po stranách vyskakovacího okna posuvníky a přesahující prvky formuláře skryl.

Ukládání států dělá serveru mírné potíže. Při vybrání všech států, kterých je asi 300, trvá ukládání o několik sekund déle než u jiných operací, protože databázový server musí do vazební tabulky přidat všech 300 záznamů. Pokud se na situaci podíváme z uživatelského

ského hlediska, uživatel si dopravu nastaví jednou za celou existenci obchodu a vícekrát už nastavení nebude upravovat.

4.3 Vytváření a úprava produktů

Produkt je pro internetové obchody tou nejdůležitější surovinou, se kterou budou obchodovat. Pro vytváření, mazání a úpravu produktů je vytvořen celý samostatný modul, protože produkt obsahuje velké množství nastavení. Na úvodní stránce je vypsána tabulka (datagrid) se všemi produkty (Obrázek 8).

Tabulka vypisuje nejen informace o produktu jako je název nebo kód, ale také v jaké je produkt umístěn kategorii, jakého má výrobce, jaká je jeho momentální dostupnost, jaké je množství na skladě, aktuální sleva a výsledná cena. Dotaz lze také omezit jen na aktivní položky nebo zboží aktuálně v akci nebo ve výprodeji. Toto vše vytahuje z databáze jeden SQL dotaz (viz Výpis 4). Ve Výpisu 4 je můj první největší dotaz takového rozsahu, který jsem vytvořil.

```

SELECT 'PD'.title ' ', P.'product_id' AS 'id', 'P'.active_yn ' ',
CONCAT(IFNULL('price_with_vat',0),',',currency_id) AS 'price', IFNULL('M'.name, '—') AS '
manufacturer_name',
IFNULL('AD'.text, '—') AS 'availability ', IF ('P'.code=' ', '—',P'.code') AS 'code', IF ('
stock'=0, '—',stock) AS 'stock',
IFNULL(CONCAT(TRIM(TRAILING ' ' FROM (TRIM(TRAILING ' '
FROM (IF('P'.discount='0',NULL,P'.discount')))), ' %'), '—') AS 'discount',
IFNULL((SELECT IF(
(SELECT 'main_yn'
FROM 'homepage_products'
WHERE 'product_id' = 'P'.product_id ),
'Homepage',
(SELECT 'CD'.name'
FROM 'products.categories' 'AC'
INNER JOIN 'category_description' 'CD' ON 'CD'.category_id = 'AC'.category_id'
INNER JOIN 'languages' 'L' ON 'L'.language_id = 'CD'.language_id' AND 'L'.default_yn = 1
WHERE 'AC'.product_id = 'P'.product_id' AND 'AC'.main_yn = 1))), '—') AS 'product_category'
FROM 'products' 'P' 15
INNER JOIN 'products_descriptions' 'PD' ON 'P'.product_id = 'PD'.product_id'
LEFT JOIN 'products_prices_calculate' 'PPC' ON 'P'.product_id = 'PPC'.product_id' AND
'PPC'.language_id = 'PD'.language_id'
LEFT JOIN 'manufacturers' 'M' ON 'P'.manufacturer_id = 'M'.manufacturer_id'
LEFT JOIN 'availabilities' 'A' ON 'P'.availability_id = 'A'.availability_id '
LEFT JOIN 'availability_descriptions' 'AD' ON 'A'.availability_id = 'AD'.availability_id ' AND
'AD'.language_id = 'en'
WHERE (('PD'.product_id = 5 AND 'PD'.language_id = 'en') OR ('PD'.product_id = 7 AND
'PD'.language_id = 'en') OR ('PD'.product_id = 8 AND 'PD'.language_id = 'en') OR
('PD'.product_id = 13 AND 'PD'.language_id = 'en') OR ('PD'.product_id = 16 AND
'PD'.language_id = 'en') OR ('PD'.product_id = 18 AND 'PD'.language_id = 'en') OR
('PD'.product_id = 17 AND 'PD'.language_id = 'cz') OR ('PD'.product_id = 9 AND
'PD'.language_id = 'de') OR ('PD'.product_id = 21 AND 'PD'.language_id = 'de'))
ORDER BY 'PD'.title, 'P'.code

```

Výpis 4: SQL dotaz pro výpis všech produktů do tabulky

Novinka	<input type="checkbox"/>	od	<input type="text"/>	do	22.04.2011	✗
Akce	<input checked="" type="checkbox"/>	od	<input type="text"/>	do	<input type="text"/>	✓
Výprodej	<input type="checkbox"/>	od	25.04.2011	do	<input type="text"/>	✓

Obrázek 9: Vstupní políčka pro datum s indikací platnosti

4.3.1 Hlavní nastavení produktu

Detail produktu má, jak už jsem zmínil, velké množství nastavení. Jako první se zobrazuje nabídka „Hlavní“. Nabídka obsahuje všeobecné informace o produktu. Zajímavostí je, že uživatel si může označit jen jazyk, ve kterém bude mít produkt vyplněn popisky. Podle zaškrtnutí se tyto popisky zobrazují. Zobrazování je uděláno jQuery metodami „fadeOut()“ a „fadeIn()“. Ty vytvoří na prvku při schovávání animaci postupného zmizení. V opačném případě vytvoří animaci postupného zobrazení.

K hlavnímu nastavení patří i políčka pro vložení data, od kdy do kdy se jedná o novinku, akci nebo zboží ve výprodeji. Správnost data ve formátu „DD.MM.RRRR“ se kontroluje regulárním výrazem (viz Výpis 5).

```
#^(0[1-9]|[12][0-9]|3[01])\.{1}(0[1-9]|1[0-2])\.{1}((19|20)[0-9]{2})\.{1}##
```

Výpis 5: Regulární výraz pro kontrolu data




Pokud formát data vyhovuje, je datum uloženo do databáze. Po uložení a znovu načtení stránky se vedle políček zobrazí zelená fajfka (Obrázek 9), pokud datum vyhovuje vůči dnešnímu datu. V opačném případě se zobrazí červený křížek (Obrázek 9), indikující neplatnost data vůči dnešnímu datu.

4.3.2 Benefity

Další položka v menu, na které jsem se podílel, byla Benefity. Je to hodně podobný modul ostatním, ale má své jisté odlišnosti. První odlišností je vypsání úplně všech benefitů v systému. U ostatních modulů se vypisují jen položky patřící k modulu. Takže pokud někdo benefit vymaže, vymaže se z celého systému, když naopak benefit přidá, přidá se ke všem produktům, jen nebude vybrán.

Jednotlivé položky lze také řadit, to se však promítne do celého systému. Řazení je řešeno AJAXově. V případě, kdy uživatel uchopí položku a přesune ji v seznamu, odešle se AJAXem seznam identifikačních čísel benefitu oddělených středníkem na konkrétní PHP skript. Ten zpracovává požadavek způsobem, který postupně prochází celý seznam a u každé položky spustí SQL dotaz pro aktualizaci. Cyklus si v paměti uchovává číslo řádku, které uloží spolu s dotazem. Toto číslo (proměnou) po aktualizaci řádku inkrementuje.

Výběr benefitu je také řešen AJAXem. JavaScriptová metoda se spouští po kliknutí na zaškrťovací políčko. Metoda si nejdřív zkontroluje, jestli je zaškrťovací políčko zaškrtnuto, podle zaškrtnutí se rozhodne, jestli přidá nebo ubere k řádku kaskádovou třídu se styly

Zobrazit	Variant	Kód	Obrázek	Dostupnost / Sklad	Finanční rozdíl	Poznámka k dostupnosti
<input type="checkbox"/> Zobrazit: <input checked="" type="checkbox"/> To basket:	Barva: červená Rozměry: šířka Length: meter	svet <input checked="" type="checkbox"/>		available 22	- 67.00 DEF - 65.00 CZK + 64.00 EUR	velký svět <input type="text"/> <input type="text"/>
<input type="checkbox"/> Zobrazit: <input type="checkbox"/> To basket:	Barva: červená Rozměry: výška Length: meter	jahoda <input checked="" type="checkbox"/>		not available 98	- 66.00 DEF	re <input type="text"/>
<input type="checkbox"/> Zobrazit: <input checked="" type="checkbox"/> To basket:	Barva: modrá Rozměry: šířka Length: meter	jahoda <input type="checkbox"/>		to 10 days 22	+ 98.00 DEF	maly <input type="text"/>

Obrázek 10: Tabulka pro editaci jednotlivých variant

a nakonec odešle AJAXový požadavek serveru. V informaci se odesílá identifikační číslo produktu, identifikační číslo benefitu a zda je benefit označen. Na základě těchto informací pak PHP skript provede přidání nebo smazání záznamu z vazební tabulky mezi produktem a benefitem.

4.3.3 Varianty

Modul „Varianty“ je částečně založen na modulu „Parametry“. Parametry slouží v systému k vytváření a úpravě parametrů (vlastností) produktu, bohužel nemohu popsat bližší podrobnosti, jak modul funguje, protože jej dělal kolega. Modul varianty (Obrázek 10) usnadňuje tvorbu variant produktů. Je tvořen velkou tabulkou, ta se chová jako obrovský formulář. Každý řádek tabulky zobrazuje jednu variantu produktu. Variantě lze nastavit, zda se bude zobrazovat, zda ji lze vložit do košíku, jaký bude mít kód, jaký bude mít obrázek, její dostupnost a množství skladem, rozdíl v ceně pro každý jazyk a poznámku zase různou pro každý jazyk.

Celá stránka je jeden velký formulář. Každá varianta má pro každý jazyk skryté položky pro úpravu finančního rozdílu a pro poznámku. Finanční rozdíl nebo poznámka se zobrazí až po kliknutí na tlačítko s ikonkou tužky. Položky jsou schovány z důvodu přehlednosti. Vidět jsou jen popisky ve výchozím jazyce. Po kliknutí na ikonku Úpravy se pomocí JavaScriptu zobrazí seznam s položkami ve všech jazycích. Zobrazené editační políčka lze vidět na obrázku 10, kde v prvním řádku jsou editační pole zobrazeny, na ostatních jsou skryty.

Zajímavou funkcí u toho modulu je kontrola kódu varianty. Kód varianty totiž nemůže být v systému dvakrát stejný. Nemůže se schodovat ani s kódem produktu. Kontrola kódu je řešena JavaScriptem a AJAXem. V případě, kdy uživatel najede na políčko s kódem skryje se potvrzovací zelená fajfka, která znamená, že je kód v pořádku. Po napsání kódu a opuštění políčka se nejdříve zkontrolují všechny kódy na stránce navzájem, v případě kdy se kódy na stránce neshodují, pošle se AJAXový dotaz na PHP skript, který se postará o kontrolu kódu v databázi. Když nebude kód nalezen v tabulce s variantami ani

The image shows a web form with the following elements:

- Nový**: A checkbox labeled "ano".
- Length**: Three checkboxes labeled "meter", "kilometer", and "decimetr".
- Barva**: Four checkboxes labeled "červená", "modrá", "zelená", and "šedá".
- Rozměry**: Three checkboxes labeled "šířka", "výška", and "délka".
- dalsji**: Four checkboxes labeled "hodnota", "hodnota2", "neco", and "Test".

Obrázek 11: Zaškrťovací políčka pro generování variant

v tabulce s produkty, ukáže se u políčka znovu zelená fajfka (viz Obrázek 10). Pro celou kontrolu je zde ještě použit skrytý formulářový prvek – hidden. Do něj jsou v průběhu kontrolování ukládány informace o validaci kódu. Při novém načtení stránky se do tohoto skrytého prvku uloží hodnota „0“. „Nula“ indikuje nezkontrolovaný kód. PHP skript po odeslání formuláře ví, že prvek nebyl žádným způsobem zkontrolován. Hodnota „1“ skrytého prvku indikuje zkontrolovaný prvek, avšak nevyhovující unikátnosti. Nakonec hodnota „2“ indikuje zkontrolovaný prvek kódu a zároveň vyhovující všem podmínkám unikátnosti kódu. Čísla „0“, „1“ a „2“ jsou do skrytého prvku vepisována JavaScriptem po JavaScriptové a AJAXové kontrole.

Nejsložitější částí při tvorbě modulu bylo generování variant. Varianty se tvoří na základě parametrů vytvořených v nastavení systému. Parametry s jejichmi hodnotami jsou vidět na obrázku 11. Každý parametr má několik hodnot. Varianty se vytvářejí na základě těchto hodnot. Je potřeba zkombinovat všechny hodnoty parametru s hodnotami následujících parametrů. Nelze však kombinovat hodnoty uvnitř jednoho parametru navzájem.

Protože může být dynamický počet parametrů, jsou dvě možnosti jak hodnoty parametrů zkombinovat. První možnost by byla dynamický počet cyklů, které by procházely postupně všechny parametry a v nich všechny hodnoty. Cykly by musely rekurzivně volat samy sebe a tak postupně projít všechny parametry. Druhá, mnou realizována, možnost funguje na principu zápisu do dvourozměrného pole (tabulky). V tabulce (viz tabulka 1) bude tolik sloupců, kolik je parametrů. Nejdříve se vypočítá počet řádků v tabulce. Pak se postupně zapisují hodnoty do tabulky pro každou hodnotu parametru stejným způsobem jako by se vytvářela tabulka pro pravdivostní hodnoty (viz tabulka 1) Po naplnění tabulky daty je třeba jednotlivé řádky uložit do databáze. Databáze obsahuje dvě tabulky pro ukládání variant: tabulku „variants“, která obsahuje data o variantách a tabulku „variants_parameters“, která je vazební tabulka mezi variantou a parametry. Pro každý řádek tabulky s variantami se nejdříve musí vytvořit SQL dotaz pro vytvoření řádku v tabulce „variants“ a pro každou buňku tabulky vytvořit SQL dotaz pro vytvoření řádku v tabulce „variants_parameters“, kam se uvádí cizí klíč jako odkaz na variantu, cizí klíč jako odkaz na parametr a cizí klíč jako odkaz na hodnotu parametru. Takže nakonec

	Length	Barva	Rozměry
1. varianta:	meter	červená	šířka
2. varianta:	meter	červená	výška
3. varianta:	meter	modrá	šířka
4. varianta:	meter	modrá	výška

Tabulka 1: Příklad tabulky s variantami produktů

Kód	Produkt	en	cz	de	Akce
6	Další produkt				
KOF-C	Kofola citrus				
9	Kalendář DE				
f5221	Ferrari Etna				

Obrázek 12: Tabulka souvisejících produktů s posuvníky

to vypadá tak, že jednomu záznamu v tabulce „variants“ připadá více variant ve vazební tabulce „variants_parameters“.

4.3.4 Související produkty

Modul pro „Související produkty“ (Obrázek 12) bude vytvářet odkazy na jiné, podobné (související) produkty. Ty se budou zobrazovat v menu na „front-endu“ při výběru produktu a budou nabízet zákazníkovi podobné výrobky. Modul je tvořen obyčejnou tabulkou, ve které lze řádky uchopením řadit. U konce řádku je seznam jazyků, ve kterých výrobek popsán a na úplném konci tlačítko pro odstranění souvisejícího produktu.

Řazení je uděláno JavaScriptem a AJAXem. Při chycení myši za obrázek s šípkami lze záznam posunovat nahoru a dolů a tím ho zařadit. Při puštění tlačítka myši na místě se spustí AJAXový požadavek na PHP skript a skript provede aktualizaci pořadových čísel u souvisejících záznamů pro daný produkt. Kliknutím na název souvisejícího zboží, se uživatel přesune na daný produkt, ne na hlavní stránku produktu, ale zpět do tohoto modulu. Kliknutím na tlačítko „Zrušit“ se provede SQL dotaz, který vymaže záznam z databázové vazební tabulky.

Modul „Související“ má jako jediný více tlačítek pro přidávání. Související zboží tak lze přidat jen pro produkt, který právě upravujeme, nebo ho lze přidat oboustranně. Tlačítko „oboustranně“ přidává vazbu nejen do produktu, který právě upravujeme, ale i do produktu, který jsme tímto přidali. Takže vazba bude obousměrná. Poslední tlačítko je „Slinkovat vše“. Tímto tlačítkem se žádné produkty nepřidávají, pouze se prochází seznam zobrazených produktů a ve všech se vytvoří vazba na všechny ostatní. Takto můžeme všechny produkty navzájem propojit.

Okno pro přidávání produktů je vytvořeno pomocí jQuery metody (viz Výpis 1). Jelikož jsme potřebovali všechny produkty nějakým způsobem vyhledávat, použili jsme datagrid. Datagrid umožňuje vyhledávání a filtrování položek uvedených v něm. Bo-

Obrázek 13: Formulář pro úpravu metráže

hužel filtrování funguje jako odeslání formuláře, proto když jsem umístil tabulku do vyskakovacího okna a snažil jsem se něco vyhledat, internetová stránka se znovu načetla a jQuery okno zmizelo. Takto vůbec neměla vypadat funkčnost okna. Vytvořil jsem proto ve vyskakovacím okně iframe (HTML tag pro zanořování oken v okně prohlížeče) a do něho teprve umístil datagrid. Vyhledávání bylo poté plně funkční.

Přidávání produktů z datagridu a jeho zobrazení v tabulce se souvisejícími produkty je uděláno složitěji JavaScriptem, přes události dokumentu. Na stránce se souvisejícími produkty je vytvořena jQuery událost „bind()“, která bude čekat na vyvolání jiným skriptem. Ve vyskakovacím okně v datagridu, po kliknutí na tlačítko „Přidat“ se vyvolá jQuery událost nadřazeného okna a řádek s produktem v datagridu se jQuery animací schová. Vyvolaná událost provede vložení produktu do databázové tabulky se souvisejícími výrobky. Nově přidané produkty se řadí na začátek. Toto jsem vyřešil jednoduchým způsobem. Nově přidaný produkt z datagridu vyskakovacího okna má pořadové číslo nula. Po vložení záznamu do vazební tabulky s pořadovým číslem nula, pošlu na databázi aktualizací dotaz, který aktualizuje všechny záznamy a každému záznamu zvýší pořadové číslo o jedničku. Tím je docíleno, že nově přidaný záznam bude vždy první v pořadí.

4.3.5 Metráž

Modul (Obrázek 13) je určený pro produkty prodávané jen v určitém rozmezí jednotek. Pokud bude zákazník prodávat například koberce, nelze koupit koberec o velikosti $1m^2$, když minimální velikost, kterou je možné si objednat, je $3m^2$. To stejné omezení platí i pro maximální velikost. Samozřejmě se nemusí jednat jen o metry, ale klient si může jednotku vybrat ze seznamu. Seznam jednotek je brán ze seznamu jednotek z nastavení systému. Jednotky jsou vybírány ve výchozím jazyce. Dále modul obsahuje pole pro popis metráže. Pole pro popisy se zobrazují jen v jazycích, které jsou aktivní pro editovaný produkt (jazyky jsou zatrženy v hlavní nabídce).

5 Teoretické a praktické znalosti

Využíval jsem znalostí z předmětů „Programovací jazyky“, ve kterých jsem se naučil objektově programovat. Framework Nette [4] je celý založen na objektech a jednotlivé moduly kategorií v systému jsou stromově uspořádány. Moduly vždy dědí vlastnosti z nadřazeného modulu. Využil jsem i znalostí z předmětu „Skriptovací a programovací jazyky“, kde jsem se naučil co je to MVC (Model View Controller) datový model. Framework Nette usnadňuje implementaci toho datového modelu a v systému je používán.

Dále mi pomohly znalosti z databázových předmětů, jako je „Oracle Academie“ a „Teorie Zpracování Dat“. Znalosti jsem použil při návrhu, vytváření a modifikaci databázových tabulek. Každý modul komunikuje minimálně se dvěma databázovými tabulkami a tabulky jsou mezi sebou vzájemně propojeny.

Zatím je systém firmy EVici ve stádiu vývoje, ale jakmile se přesune do stádia testování, chci využít svých znalostí z předmětu „Tvorba informačních systémů“, kdy bude potřeba aplikaci otestovat nad velkým počtem dat v databázi (řádově stovky tisíc). Budeme testovat především dotazy, které by mohly nebezpečně zpomalovat komunikaci s databázovým serverem, protože spojují více než deset databázových tabulek.

6 Scházející znalosti a dovednosti

Firma EVici webdesign s.r.o. [1] používá pro vývoj svého systému framework Nette [4]. Naučit se základy tohoto frameworku byla nezbytná podmínka pro vykonávání odborné praxe ve firmě. Základy jsem se učil přibližně dva týdny, během kterých jsem si zkusil vytvořit malý redakční systém. Redakční systém uměl pouze vytvářet, upravovat a mazat články a řadit je do vybraných kategorií.

Další znalost, která mi chyběla, byla znalost technologií JSON a AJAX. I když veškerou asynchronní komunikaci zprostředkovává framework, znalost těchto technologií jsem potřeboval pro ladění systému a odchyťování chyb způsobených špatně napsaným kódem.

7 Dosažené výsledky

Výsledkem mé práce byl vždy jeden plně funkční modul. Pokud bylo potřeba modul upravit, pokaždé jsem ho podle potřeb upravil, aby splňoval zadané požadavky a byl zase plně funkční a použitelný.

Praxe ve firmě byla pro mě velikým přínosem, protože jsem si začal uvědomovat spojitosti s teoretickými znalostmi. Pochopil jsem, co je to vývoj aplikace, kde nestačí jen mít nápad, ale je potřeba schopných lidí, kteří společnými silami projekt dokončí. Taky jsem se dozvěděl, že vytvoření a dokončení projektu není jasná výhra. Projekt je nejdříve potřeba řádně otestovat, odstranit všechny chyby, pak nastupuje na řadu marketing a obchodní zkušenosti, kde se projekt předvádí klientele a nemalou roli hraje také technická podpora, která pomáhá klientům, když si nevědí rady.

U větších projektů, jako je například tento od firmy EVici [1], je vhodné použití nástroje pro organizaci času, protože ve změti papíru a poznámek se člověk snadno ztratí. Firma využívá takový online workflow program firmy WORKetc. Já jsem však nebyl do tohoto programu zahrnut a práce mi byla zadávána ústně nebo ve formě emailu.

8 Reference

- [1] *EVici webdesign s.r.o.* [online]. 2003 [cit. 2011-04-26]. Dostupné z WWW: <www.evici.cz>.
- [2] *Wikipedia* [online]. 1995 [cit. 2011-04-26]. Dostupné z WWW: <www.wikipedia.org>.
- [3] *PHP: Hypertext Preprocessor* [online]. 2001 [cit. 2011-04-26]. Dostupné z WWW: <www.php.net>.
- [4] GRUDL, David. *Nette Framework* [online]. 2001 [cit. 2011-04-26]. Dostupné z WWW: <www.nette.org>.
- [5] Nette Foundation. *dibi* [online]. 2001 [cit. 2011-04-26]. Dostupné z WWW: <www.dibiphp.com>.
- [6] *jQuery: The Write Less, Do More, JavaScript Library* [online]. 2010 [cit. 2011-04-26]. Dostupné z WWW: <www.jquery.com>.
- [7] *jQuery návod* [online]. 2010 [cit. 2011-04-26]. Dostupné z WWW: <www.jquery-navod.cz>.
- [8] ODVÁRKO Jan. *JSColor - JavaScript / HTML Color Picker, Selector, Chooser* [online]. 2008 [cit. 2011-04-26]. Dostupné z WWW: <www.jscolor.com>.